NASA Technical Memorandum 81291

USAAVRADCOM TR-81-A-12

(NASA-TM-81291) LOGIC ANALYSIS OF COMPLEX SYSTEMS BY CHARACTERIZING FAILURE PHENOMENA TO ACHIEVE DIAGNOSIS AND FAULT-ISOLATION (NASA) 22 p HC A02/MF A01 CSCL 14D

N81-26467

Unclas

G3/38 26732

Logic Analysis of Complex Systems by Characterizing Failure Phenomena to Achieve Diagnosis and Fault-Isolation

James T. Wong and William L. Andre

April 1981

NASA

[illegible]

United States Army
Aviation Research
and Development
Command

US ARMY
AVRADCOM

NASA Technical Memorandum 81291 USAAVRADCOM TR 81 A 12

Logic Analysis of Complex Systems by Characterizing Failure Phenomena to Achieve Diagnosis and Fault-Isolation

James T Wong

and

William L. Andre, U.S. Army Research and Technology Laboratories (AVRADCOM)
Ames Research Center, Moffett Field, California

NASA
[illegible]
[illegible]

Ames Research Center
[illegible]

United States Army
Aviation Research and
Development Command
St Louis Missouri 63166

SUMMARY

It is often difficult if not impossible to analyze a design of a large, complex system with respect to its reliability parameters and maintenance characteristics when numerous elements are functionally interdependent. A recent result has shown that, for a certain class of systems, the interdependency among the elements of such a system together with the elements constitutes a mathematical structure — a partially ordered set. It is called a loop-free logic model of the system. On the basis of an intrinsic property of the mathematical structure, a characterization of system component failure in terms of maximal subsets of bad test signals of the system was obtained. Also, as a consequence, information concerning the total number of failure components in the system was deduced. Detailed examples are given to show how to restructure real systems containing loops into loop-free models for which the result is applicable.

INTRODUCTION

Availability is a system parameter and is used to measure the operational readiness of a system/equipment by the reliability and maintainability engineering community. It is defined as the ratio of mean time to failure (MTTF) to the sum of mean time to repair (MTTR) and MTTF. According to the definition, availability is a child of the marriage between these two disciplines. The theory of reliability deals with a system's ability to perform its intended function under a prescribed condition for a period of time without failure. Maintainability concerns itself with a system's ability to restore a system to its operational condition or to prevent unnecessary failure. In both instances, time is the important common factor used to measure the "up" condition (free from failure) on the one hand and the "down" condition (by failure) on the other.

One way to increase system availability is to reduce the MTTR which partly hinges on the ability to correctly prognosticate the state of the system and to identify the failed components if the system is malfunctioning.

Recently, the U.S. Army Research and Technologies at Moffett Field, California, established a mathematical basis for complex systems without loops — called a logic model theory. In the theory it has been established (ref. 1) that the minimum number of test points required for conclusive detection of system failure is equal to the total number of terminal test points; this set of points constitutes the optimal choice. This result is useful for system checkout or prognosis. On the basis of the theory, we have etablished some results whose application is complementary to that of the foregoing one. In particular, it is shown that every maximal subset of bad events of the system corresponds to a failed component, and the converse of this statement is also true if a further assumption is imposed. Also, as a consequence, it has been deduced that the total number of failures is at least as many as the total number of maximal subsets of bad events.

ASSUMPTIONS AND DEFINITIONS

In this section, we shall state explicitly the assumptions and definitions upon which the following development is based. It is hypothesized that:

1. *At any instant or stage, the system or equipment under consideration may be in only one of two states: functioning or faulty.*

2. *The system can be schematically decomposed into a finite number of components (or modules), each of which, at any instant, is in one of the two possible states.*

3. *The state of the system depends solely on the states of its components.*

4. *The system is loop-free.*

Hypothesis (1) is a realistic assumption because, if the performance level of a given component is degraded to an "unsatisfactory" level (or beyond the tolerance as specified in the specifications of the equipment), then the component is in the malfunctioning state. Assumption (2) demands only the feasibility of schematic system decomposition, not necessarily a physical decomposition. Also, it is tacitly assumed in (3) that a proper environment exists for the system under question. This also implies that input to the system is considered to be good. Assumption (4) imposes a limitation on the applicability of this study to systems containing functional dependence loops. For those cases, one should restructure each functional loop as a component and an event. Hence the resulting model would be loop-free.

In addition to the above assumptions, one needs the following.

$C \equiv$ nonempty finite set of all components of a given system.

$E \equiv$ nonempty finite set of all events (signals) of the system.

$S \equiv$ set of all functional entities, defined to be the union of C and E.

$P \equiv$ set of all peripheral components.

$U \equiv C \cup P$ or $P \cup C$

A partially ordered set P is a structure consisting of a set S and a relation $\sim$ satisfying the following postulates:

1. (Reflexive): $a \sim b$ and $b \sim a$ hold if and only if $a = b$, where $a,b \in S$.

2. (Transitive): $a \sim b$ and $b \sim c$ imply $a \sim c$, where $a,b,c \in S$.

Let $\alpha,\beta \in S$. Then α depends functionally on β (symbolically, $\alpha < \beta$) if there exist functional entities $\ell_1, \ell_2, \ldots, \ell_n$ such that $\alpha < \ell_1 < \ell_2 < \ldots < \ell_n < \beta$.

An event (or signal) is said to be bad if it is out of specification; otherwise, it is said to be good. A component in a loop-free system is said to be malfunctioning if all its input signals (events) are good and at least one of its output signals (events) is bad; it is said to be functioning if all its outputs are good.

$\Lambda_a(O_j) \equiv \{s \in E \mid s < O_j\}$, O_j is an output of a, $a \in C$.

$\bar{\Lambda}_a(O_j) \equiv$ all events (signals) in $\Lambda_a(O_j)$ are bad.

Let $a \in C$ and O_j be an output of a. Then a set of bad events (signals), $\bar{\Lambda}_a(O_j)$, is said to be maximal if for all $b \in U$, $\bar{\Lambda}_a(O_j) \subset \bar{\Lambda}_b(O_i)$ implies $\bar{\Lambda}_a(O_j) = \bar{\Lambda}_b(O_i)$.

Now we state the last assumption, which allows for flexibility in modeling at different levels.

5. *For every event* $s \in E$, *there is a component* $c \in U$ *having* s *as its output.*

Finally, in the interest of completeness, we define explicitly the following terms.

Component — a collection of one or more items.

Event (signal) — a measurable or observable quantity.

Functional entity — a component or an event.

Dependence — a functional relationship between two functional entities.

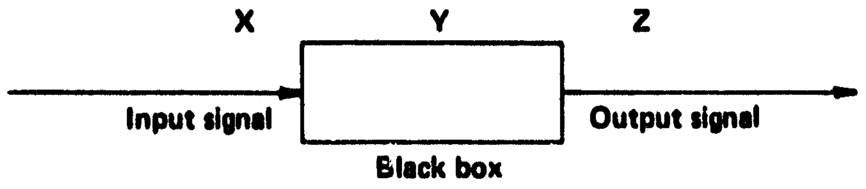
Dependency chain — a collection of two or more functional entities for which dependence exists.

Loop — a closed dependence chain.

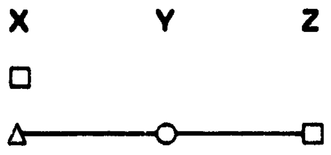
BACKGROUND

In this section we discuss the basic idea of the logic model concept together with some notions that will be required to understand the following development.

Structurally, a logic model consists of a collection of dependency chains arranged in a particular order that reflects the functional relationship that existed between the components and the observable or measurable state of nature of a system or equipment. The dependency chains are the basic building block of the logic model concept. A simple example of a dependency chain can be illustrated by the "black-box" concept as follows:



This simple input-output mechanism shows that the output signal Z depends functionally on the operational status of the black box Y (or simply the black box Y) and the input signal X. The dependency chain of this simple mechanism is pictorially represented as



where □ represents an observable or measurable state of nature, ○ represents the functional component, and Δ denotes the dependency of one functional entity on another. This symbolic representation of the dependency chain also yields a logic model of the mechanism — a logic model that consists of only one dependency chain.

A more complex example is a simple power relay circuit together with a logic model as shown in figure 1. This logic model consists of four dependency chains. Note that, for example, the signal at TP-2 depends on the operational status of the transformer T1 and the signal at TP-1. This is a dependency chain and it reflects the power transition portion on one leg of the transformer, whereas the chain corresponding to TP-3 reflects power transition for the other output leg.

The foregoing example has been generated manually and reported in reference 2. In this case, the manual generation of the model has been an easy task because there are altogether only a few functional entities in the model — five events and three components. (Note items S1 and R1 together are considered as a component.) For a more detailed modeling of a complex system, an automated capability for generating a logic model not only is desirable but becomes necessary.

CHARACTERIZATION OF COMPONENT FAILURE

The set of all functional entities S together with the functional dependence, $<$, as a relation on S constitutes a mathematical structure. In fact, as was established in a previous report (ref. 1), every loop-free logic model is a partially ordered set, and from this some interesting results that are useful for maintenance analysis were deduced.

This unique property of logic models enables us to obtain a characterization of system component failure phenomena, whose proof is given in the appendix.

Theorem 1: Let $a \in C$ *and* L *be a loop-free logic model. Then component* a *is malfunctioning if, for some output* O_j *of* a, $\bar{\Lambda}_a(O_j)$ *is a maximal subset of the set of all bad events in* E.

This pleasing result would not necessarily hold if the loop-free constraint were relaxed. For then we could have a malfunctioning component in a loop and, in this case, the logic model is not a partially ordered set. Hence the asymmetric property of a partially ordered set does not apply, and it might lead to an incorrect identification of failed components. This undesirable feature on the applicability can be circumvented to a degree, provided a certain degradation on the logic model is admitted. A detailed dscussion along with some examples follows in the next section.

The converse of the Theorem is generally not true. So it is somewhat unorthodox to call it a characterization because such usage implies an equivalency of two statements in the mathematical literature. However, we do have a weaker equivalence result.

Theorem 2: Suppose U *contains at most one malfunctioning component, and let* L *be a loop-free logic model. Then a component* $a \in C$ *is malfunctioning if and only if, for some output* O_j *of* a, $\bar{\Lambda}_a(O_j)$ *is a maximal subset of the set of all bad events in* E.

Before we continue, note that in this weaker form of characterization, an additional hypothesis is assumed — namely, we allow not more than one failure component, at a given instant, within both the system components and the peripheral components. Also, the Theorem would not be necessarily valid if the peripheral portion were omitted unless all the peripheral components were assumed to be good.

The next result states that the totality of bad events in a malfunctioning system is precisely the set theoretic sum of all the maximal subsets of bad events in the system.

Theorem 3: The set of all bad events in $\bar{\text{E}}$ *of a loop-free logic model is equal to the union of all maximal subsets* $\bar{\Lambda}_a(O_j)$ *in* E.

Finally, as a consequence of the foregoing results, we deduce the following corollary.

Corollary: If each component of a loop-free logic model has only one output, then the number of failed components of a malfunctioning system is equal to or greater than the total number of maximal subsets of bad events.

This Corollary can only conclude that there are at least as many failed components as the number of maximal subsets of bad events. Equality does not hold in general because the converse of Theorem 1 is not necessarily true.

SYSTEMS WITH LOOPS

It is not uncommon for systems to contain closed dependence chains or loops, especially in electronic equipment. As discussed under "Assumptions and Definitions," for such a system it is necessary to degenerate each loop into a component and an event so that the resulting logic model is loop-free, and for which the result obtained under "Characterization of Component Failure" is applicable.

Figure 2 is a computer-generated logic model, published in reference 2, of a radio used extensively in Army helicopters such as the UH-1. There are two loops (or closed dependency chains) in this relatively complex model, which consists of 51 dependency chains involving 175 functional entities. Concatenation of the following four dependency chains:

Event A038 depends on component I044 and on event A037.

Event A037 depends on component I043 and on event A036.

Event A036 depends on component (I042, I110) and on event A076.

Event A076 depends on component (I089, I088, 1046, I045) and on events A038, A009, A010, A011.

is a dependency chain involving a loop, as shown in figure 3. To eliminate this loop, we must remodel the events A038, A037, A036, and A076 as one event called Axxx, where xxx is unique in the remaining event set and items I044, I043, I042, and I110 as one item Iyyy or component, where yyy is unique in the remaining items of the model. The other loop in the model is embedded in the following dependency chains:

Event A070 depends on component I100 and on event A069.

Event A069 depends on component I099 and on event A060.

Event A060 depends on component I098 and on event A059.

Event A059 depends on component I097 and on event A058.

Event A058 depends on component (I096, I095) and on events A070, A040, A009, A010, A011.

as shown in figure 4. Here, as before, the events involving in the loop as well as the items must be degenerated into a unique representation. Now, the model is loop-free, and the apparent price to be paid for this action is an inability to fault isolate down to the same level if the loop did not exist.

The foregoing model involves simple loops. A more complicated example (fig. 5) of loop embedding is provided by a logic model of an on/off gating circuit board (ref. 3) used on special underwater surveillance gear by the Navy. A careful examination of the model reveals that there are five loops embedded in the following eight dependency chains:

Event A038 depends on component I030 and on events A045, A046, A027, A044.

Event A045 depends on component I038 and on events A052, A037.

Event A052 depends on component I041 and on events A034, A033.

Event A046 depends on component I015 and on events A030, A031.

Event A027 depends on component I015 and on events A030, A031.

Event A034 depends on component I023 and on events A030, A040, A037, A041.

Event A030 depends on component I017 and on events A034, A033.

Event A033 depends on component I022 and on events A038, A037, A036.

To obtain a loop-free model, we identify events A038, A045, A052, A046, A027, A034, A030, and A033 as another event that is unique among the remaining events; similarly, items I030, I017, I023, I015, I038, and I041 should be grouped as a new component with a unique identification.

CONCLUDING REMARKS

On the basis of the previously established result that every loop-free logic model is a partially ordered set, we have found that every maximal subset of bad events corresponds to a failure component. This result, together with the fact that every bad event of a malfunctioning system belongs to some maximal subset of bad events, enables us to deduce that the number of failure components is equal to or greater than the total number of maximal subsets of bad events. The equality does not generally hold because it is not necessarily true that every failed component gives rise to a maximal subset of bad events. However, this statement is true for systems containing only at most one failed component at a given instant.

APPENDIX

Theorem 1: Let L *be a loop-free model and* $a \in C$. *Then component* a *is malfunctioning if, for some output* O_j *of* a, $\bar{\Lambda}_a(O_j)$ *is a maximal subset of the set of all bad events in* E.

Proof: First we observe that $O_j \in \bar{\Lambda}_a(O_j)$, so O_j is a bad output of component a. The proof of the Theorem now is reduced to prove that all the inputs, to component a, upon which O_j depends are good. Assuming the contrary, suppose there is an input s to a that is bad. Then, by definition, there is a component $b \in U$ such that s is an output of b. Event s being bad implies that all the events in $\bar{\Lambda}_b(s)$ are bad. So, we have $\bar{\Lambda}_a(O_j) \subset \bar{\Lambda}_a(O_j) + \{s\} \subset \bar{\Lambda}_b(s)$. But $\bar{\Lambda}_a(O_j)$ is maximal implies $\bar{\Lambda}_a(O_j) = \bar{\Lambda}_b(s) = \bar{\Lambda}_a(O_j) + \{s\}$. Then it follows that $s < O_j$. Since L is a partially ordered set, we have $s = O_j$, or input is the same as output, a contradiction.

Theorem 2: Suppose U *contains at most one malfunctioning component and* L *be a loop-free logic model. Then a component* $a \in C$ *is malfunctioning if and only if, for some output* O_j *of* a, $\bar{\Lambda}_a(O_j)$ *is a maximal subset of the set of all bad events in* E.

Proof: The proof for the sufficient condition is carried over from that of Theorem 1. Now we want to prove that it is necessary also. Suppose $\bar{\Lambda}_a(O_j)$ is not maximal. Then there is a set $\bar{\Lambda}_b(s)$ such that it contains $\bar{\Lambda}_a(O_j)$ as a proper subset. Now, if $\bar{\Lambda}_b(s)$ is maximal, then b is malfunctioning, in which we have a contradiction since U contains two failure components a and b. On the other hand, if $\bar{\Lambda}_b(s)$ is not maximal, then there exists $\bar{\Lambda}_c(t)$ such that $\bar{\Lambda}_b(s) \subset \bar{\Lambda}_c(t)$. Now applying the same argument, we have either c malfunctioning or else the existence of a set $\bar{\Lambda}_d(n)$. The process will terminate eventually since the set U is finite. Thus, it leads to a contradiction.

Theorem 3: The set of all bad events in E *of a loop-free logic model is equal to the union of all maximal subsets* $\bar{\Lambda}_a(O_j)$ *in* E.

Proof: Let T denote the set of all bad test points in E, and $R = \bigcup_a \bar{\Lambda}_a$, where $\bar{\Lambda}_a = \bigcup_{O_j} \bar{\Lambda}_a(O_j)$ and $\bar{\Lambda}_a(O_j)$ is a maximal subset for some O_j. To prove the theorem, it is only necessary to prove that $T \subset R$ because obviously each element of R is also an element in T. So, let s_1 be an element in T. We want to show that s_1 is an element of some maximal subset. To begin with, by definition, there is a component c_1 having s_1 as its output. It follows that the set $\Lambda_{c_1}(s_1)$ is a subset of bad test points. Hence, if $\bar{\Lambda}_{c_1}(s_1)$ is maximal, the assertion follows; otherwise, there exists a bad test point s_2 such that $\bar{\Lambda}_{c_1}(s_1)$ is a proper subset of $\bar{\Lambda}_{c_2}(s_2)$, for some component c_2. Unless $\bar{\Lambda}_{c_2}(s_2)$ is maximal, the process can be continued and eventually terminates itself since there are only a finite number

of test points in a given logic model. Therefore,
$s_1 \in \bar{\Lambda}_{c_1}(s_1) \subset \bar{\Lambda}_{c_2}(s_2) \subset \ldots \subset \bar{\Lambda}_{c_k}(s_k)$ is maximal.

Corollary: If each component of a loop-free logic model has only one output, then the number of failed components of a malfunctioning system is equal to or greater than the total number of maximal subsets of bad events.

Proof: Let the nonnegative integer k be the number of failed components in the logic model under consideration. Then the hypothesis and Theorem 1 imply that, for each failure component, there corresponds one and only one maximal subset of bad test points. So, we have k maximal subsets of bad test signals, but Theorem 2 implies these are the only maximal subsets. Hence K cannot be less than the number of maximal subsets. On the other hand, the converse of Theorem 1 is not necessarily true. It follows that k can be greater than the total number of maximal subsets of the set of all bad test signals.

REFERENCES

1. Wong, James T.; and Andre, William L.: Some Generic Properties of a Logic Model for Analyzing Hardware Maintenance and Design Concepts. Proceedings of the Symposium on Applications of Decision Theory to Problems of Diagnosis and Repair, Fairborn, Ohio, June 2-3, 1976.

2. LOGMOD Diagnostic Test Set and Demonstrator. Prepared by DETEX Systems, Inc., for U.S. Army Research and Technology Laboratories (AVRADCOM), Ames Research Center, Moffett Field, Calif., June 1977.

3. LOGMOD Diagnostic Procedures for Mechanical and Electrical Equipment. Prepared by DETEX Systems, Inc., for U.S. Army Research and Technology Laboratories (AVRADCOM), Ames Research Center, Moffett Field, Calif., June 1975.

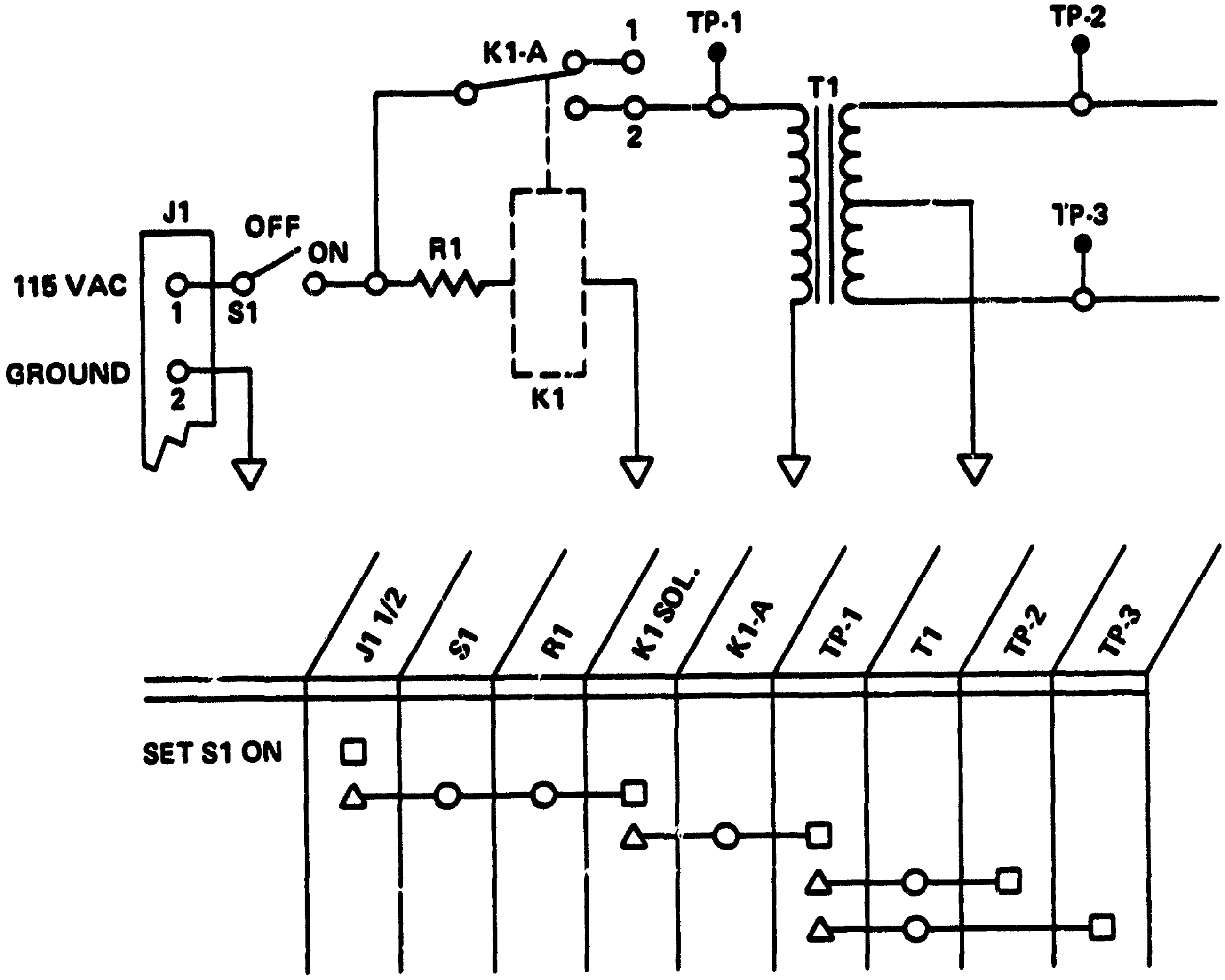


Figure 1.- A Power relay and a corresponding logic model.

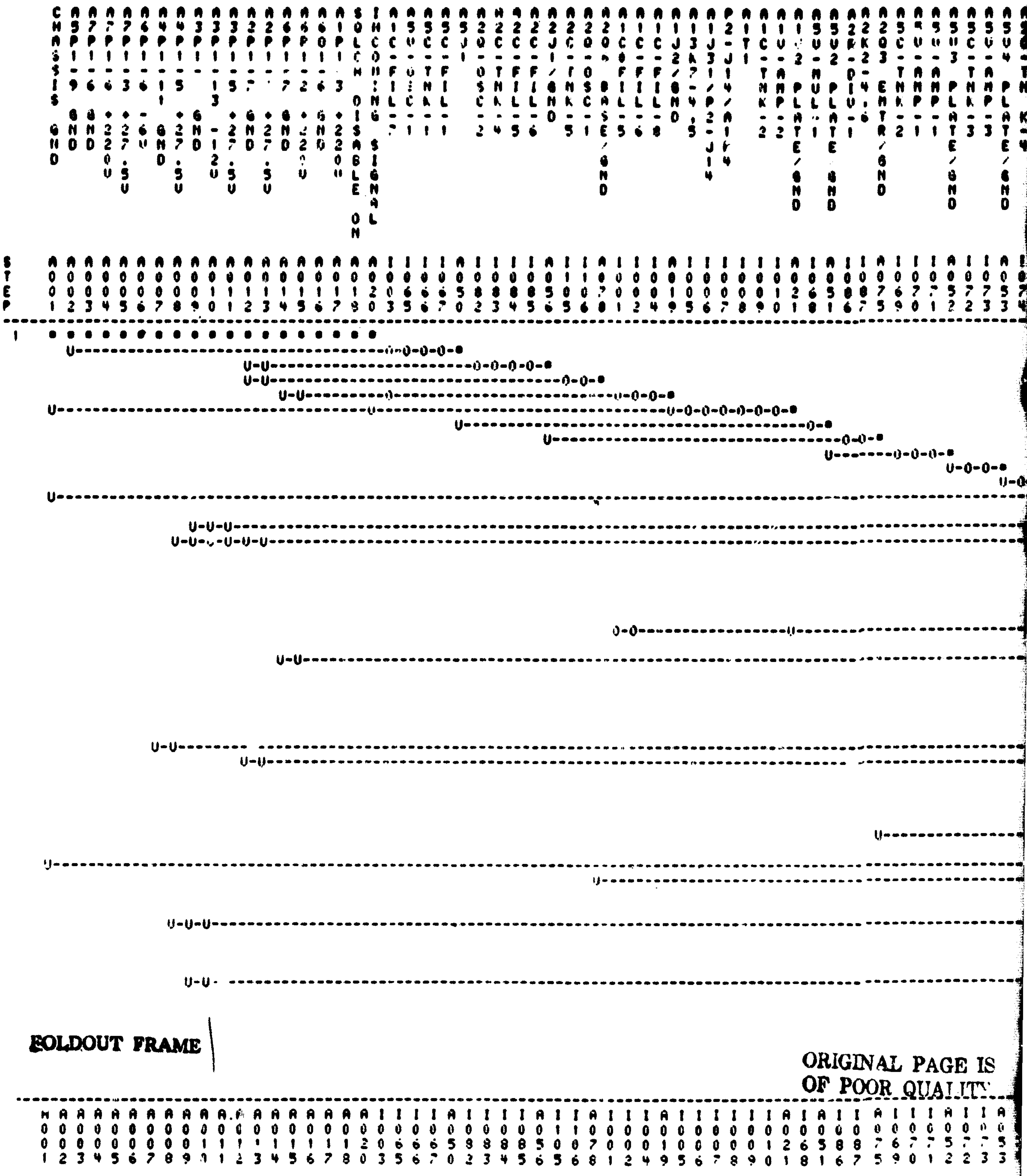
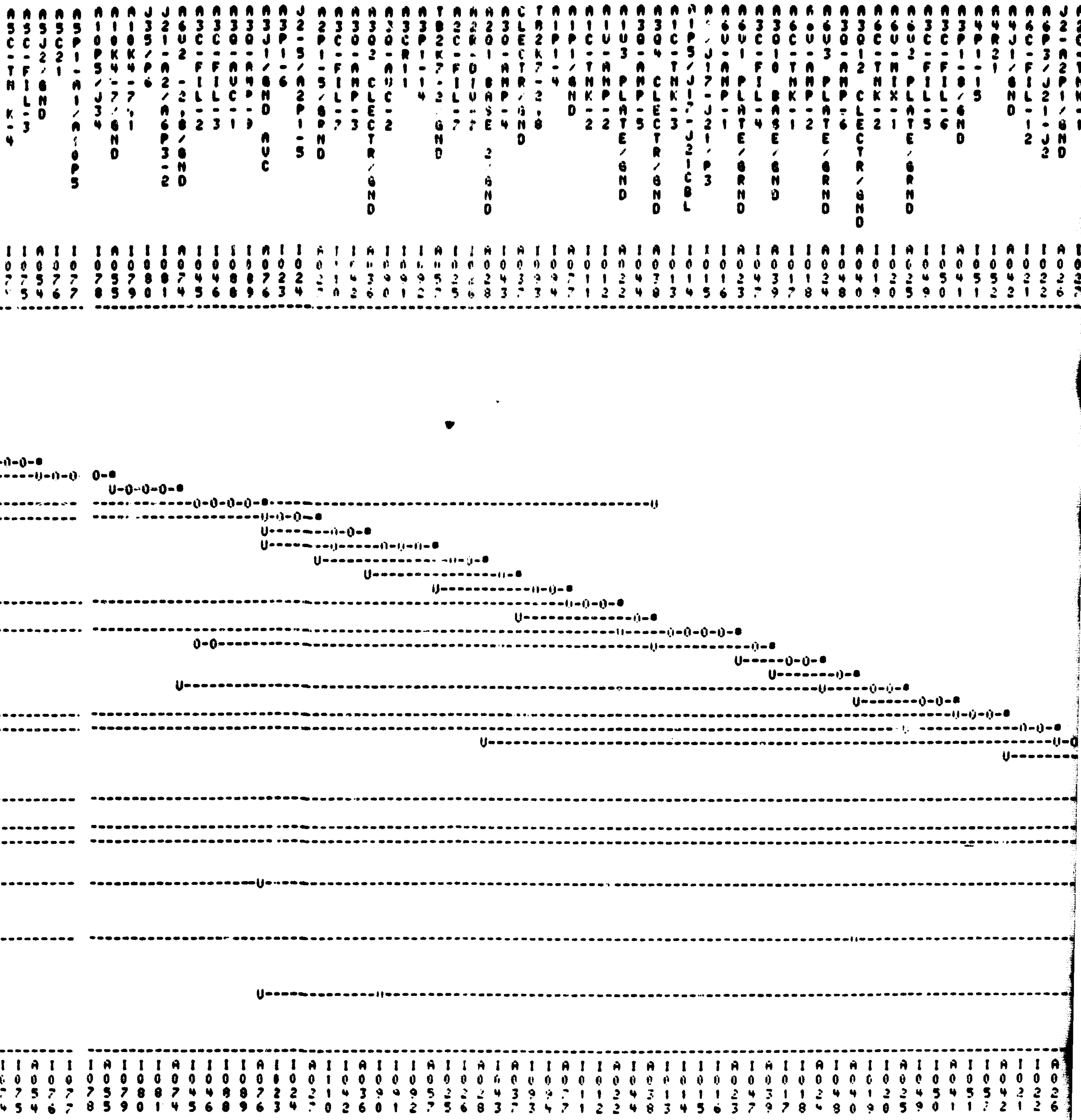
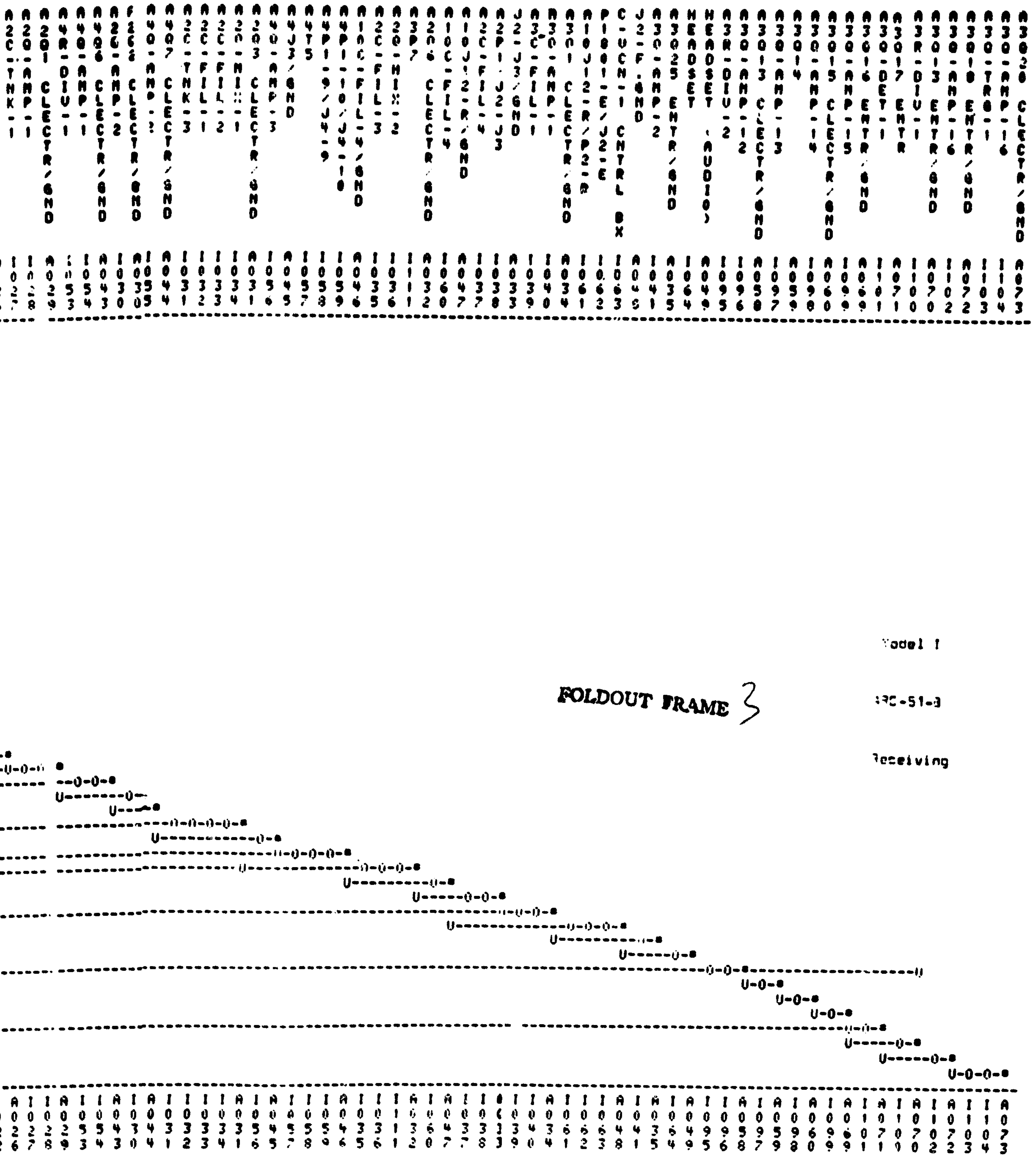


Figure 2.- Logic model with loops.



FOLDOUT FRAME

2



Model I

FOLDOUT FRAME 3

SRC-51-3

Receiving

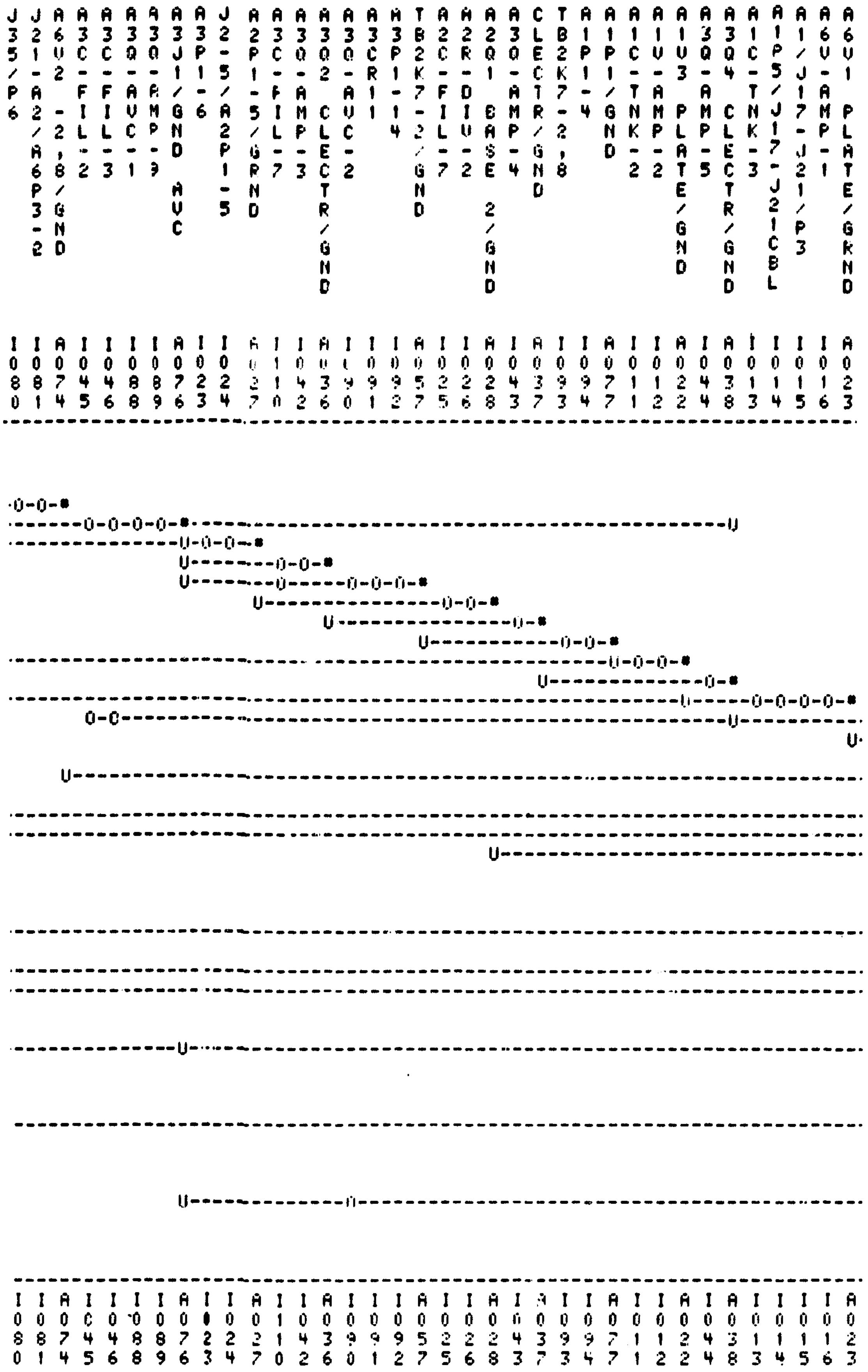


Figure 3.- Loop embedded in logic model.

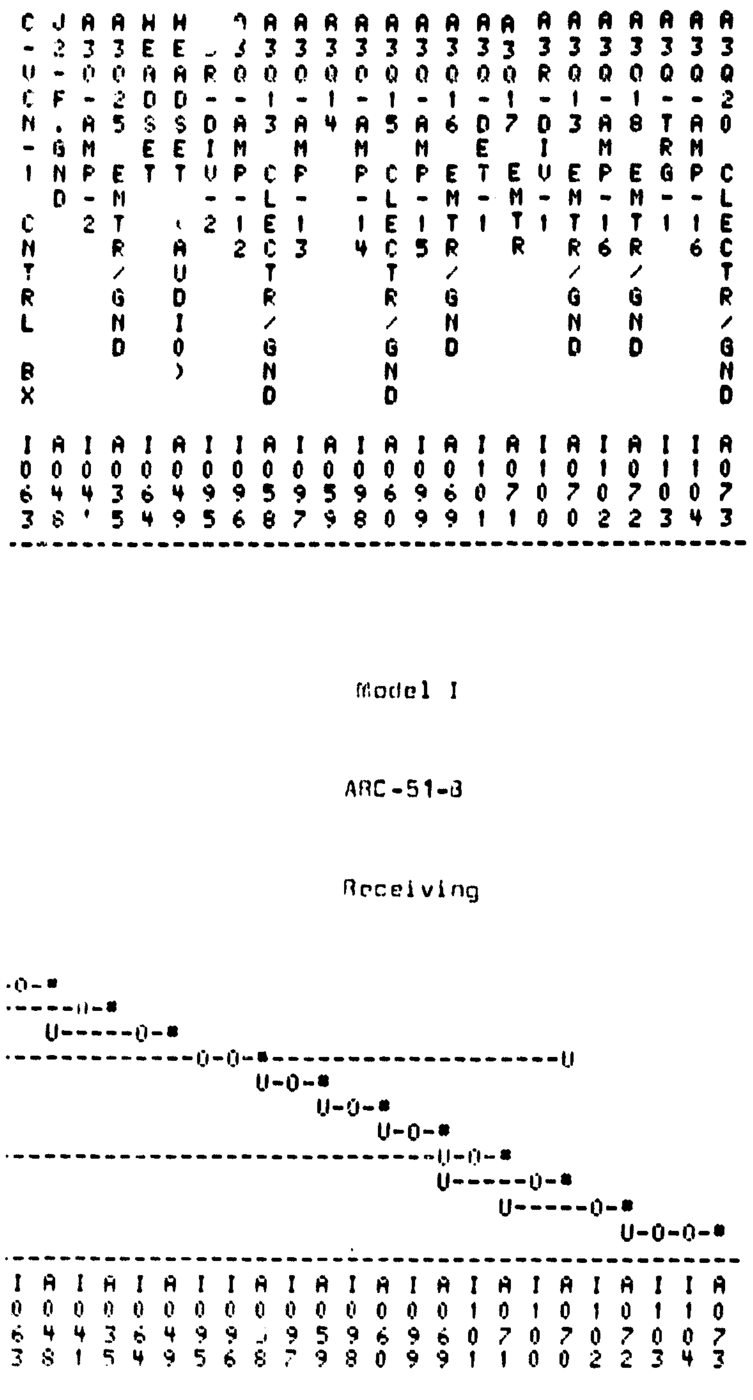


Figure 4.- Loop embedded in the model.

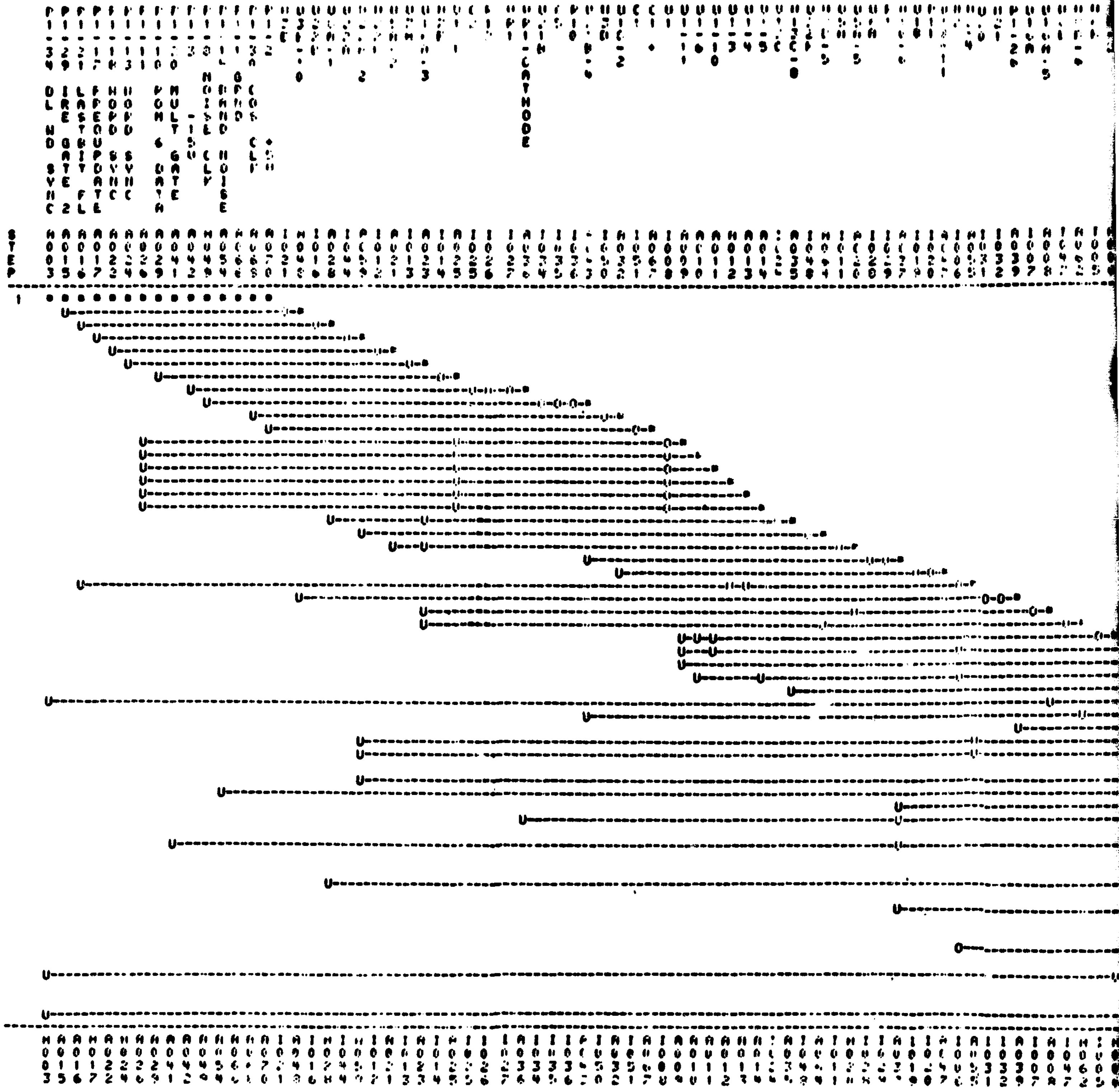
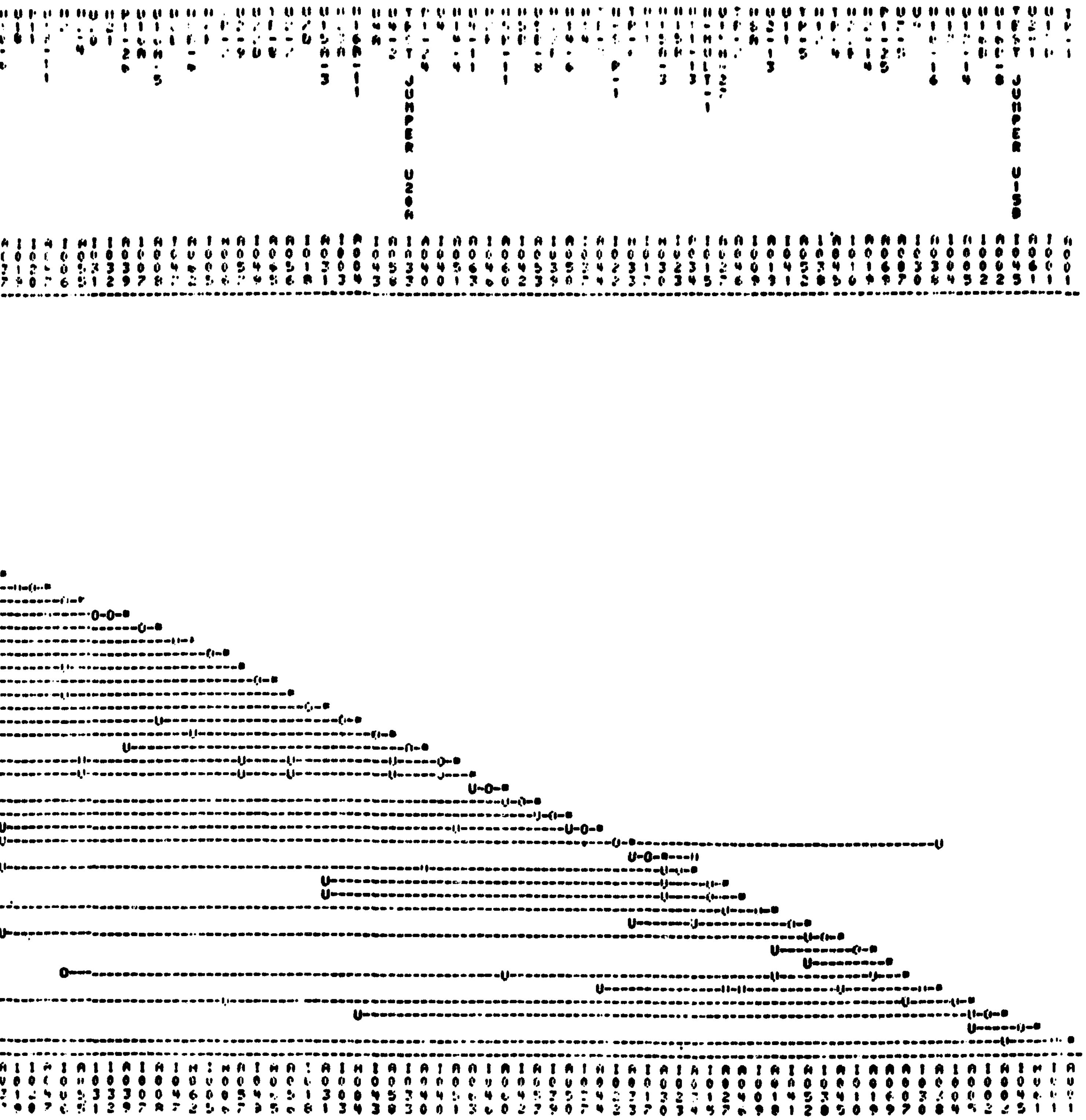


Figure 5.- Logic model with loops interwoven.



FOLDOUT FRAME 2